MOFFETT AI

Moffett Antoum®: A Deep-Sparse AI Inference System-on-Chip for Vision and Large-language Models

Dr. Zhibin Xiao & Team
Co-founder and Chief Architect
Moffett AI

Hot Chips 35, Aug 29, 2023

HOT CHIPS

Outline

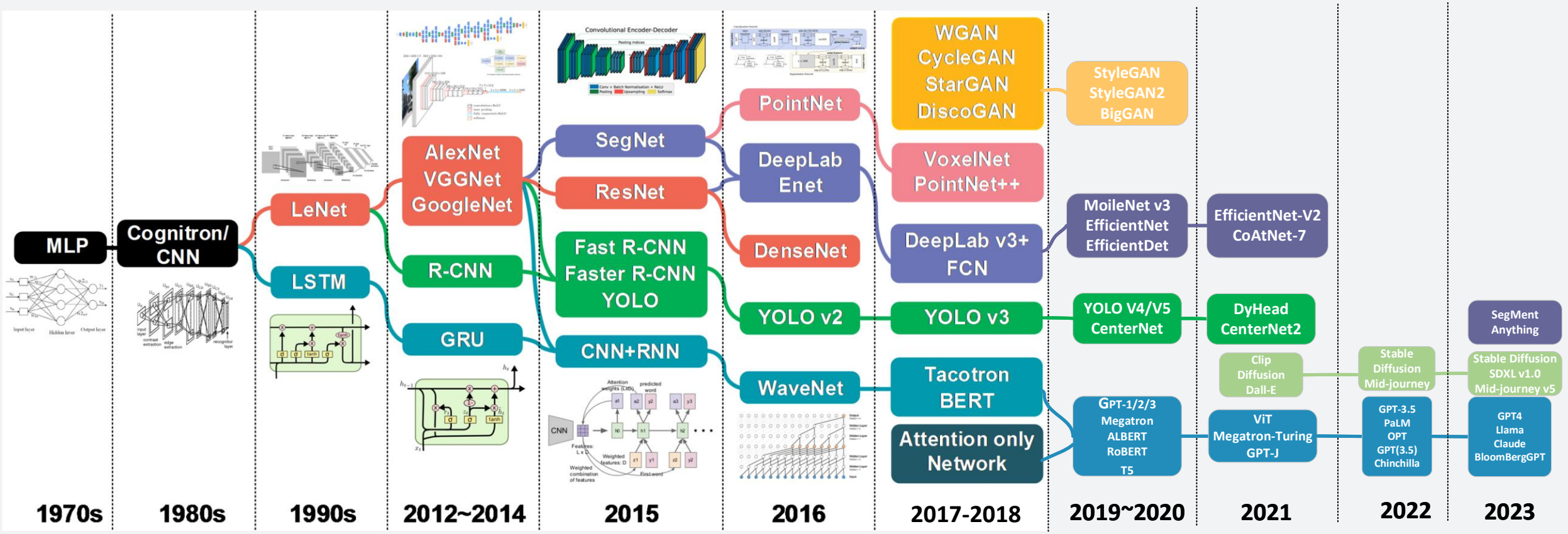
- **ML Trends and Antoum® Chip Overview**
- Sparsity in ML Inference and Design Methodology
- Antoum® Chip uArch and SparseOne Cards
- Software Toolchain
- Application Performance and Demo

A Brief History of AI Models

MOFFETT AI

Computer Vision (CV) Models Explosion

Large Language Model (LLM) Explosion



AI Model Evolution Trends

Model architecture and operator converges

Model and dataset size explodes

HOT CHIPS

* Source: Partly from **Hoi-Jun Yoo,** ISSCC 2019

Characteristics of Vision and Language Models

MOFFETT AI

Vision Models

- Small models (millions of parameters)
- Large Input Size (4k/8k images)
- Throughput-sensitive within latency constraint
- From convolution to transformer
- Non-AI functions (image/video/pre-post processing)
- Higher Parallelism and **Computation-bounded**

Language Models

- Large Models (billions of parameters)
- Small input size (context window 128 – 32K)
- Latency and Throughput Sensitive
- Data-dependent computation (token by token)
- Pre-post processing: Tokenizer, Beam Search (etc.)
- Single-card to multiple-card inference
- **Memory or I/O bounded**

Sparsity benefits both Vision and Language Models:

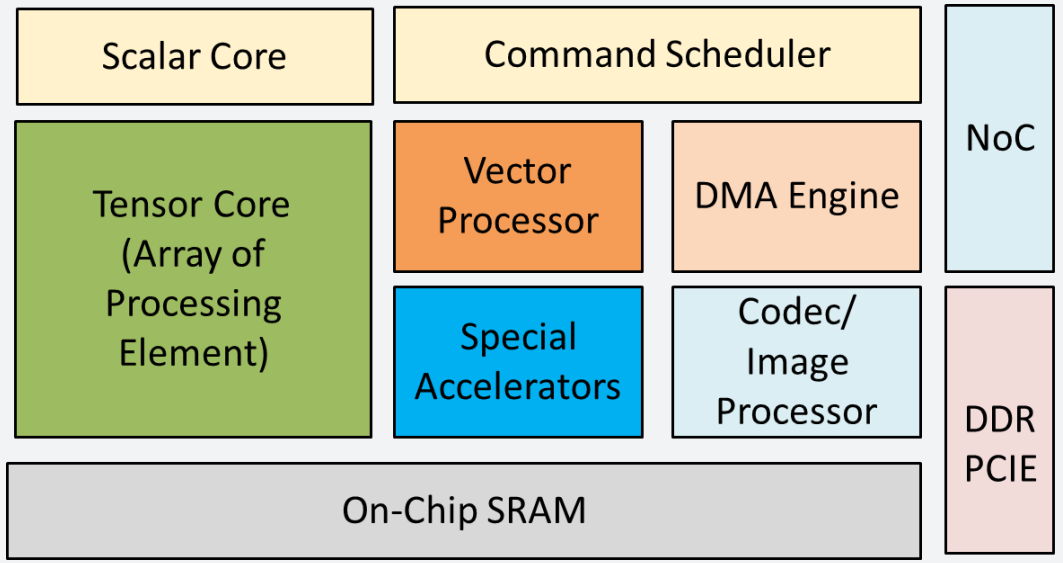
Reduce memory capacity and bandwidth requirement

Faster Computation

Antoum®: A Deep-Sparse AI Inference SoC

MOFFETT AI

- **Target Applications:**
  - **Vision Applications:** Classification, Object Detection, Segmentation, Super Resolution, etc.
  - **AIGC Applications:** Diffusion Models, Transformers
- **Key Features:**
  - Domain-specific Architecture for maximum efficiency
  - **Efficient Deep-Sparse Tensor Core (up to 32x sparsity)**
  - Programmable Vector Processor with large vector size
  - Near-memory processing with large on-chip SRAM
  - Scalable design with network-on-chip (NoC)
  - A full System-on-Chip (SoC) with general interfaces, special accelerators and application processors running Linux



A High-level Antoum® Chip Architecture

HOT CHIPS

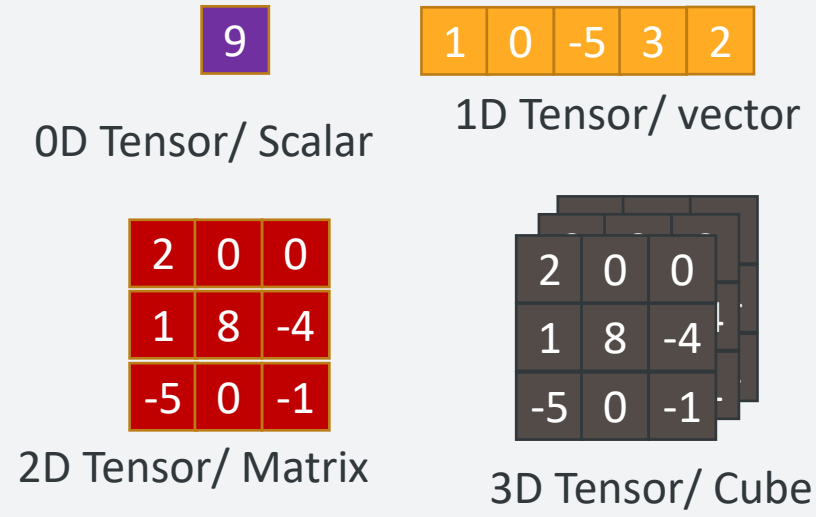
Outline

- ML Trends and Antoum® Chip Overview
- **Sparsity in ML Inference and Design Methodology**
- Antoum® Chip uArch and SparseOne Cards
- Software Toolchain
- Application Performance and Demo

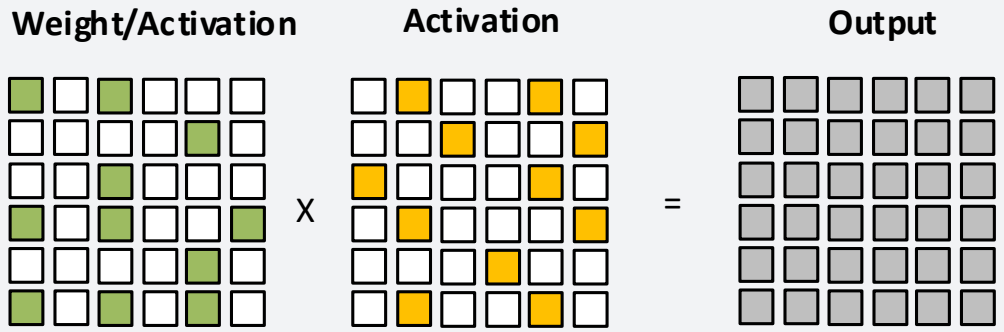
Sparsity in ML Inference

MOFFETT AI

- The core of ML inference is **Tensor Algebra**
- Zeros naturally exist or can be induced in Tensors
- No need to store zero or compute zero in a tensor
  - **Huge benefits:** less storage, computation time, memory bandwidth, reduce power
  - "Sparsity Tax": Extra HW cost for compression, decompression, schedule (limit the throughput and extra power/area overhead)



Tensor Format

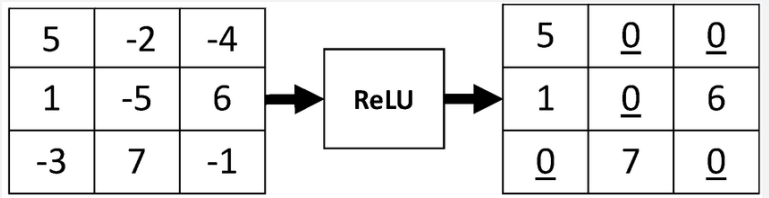


Sparse Matrix Multiplication

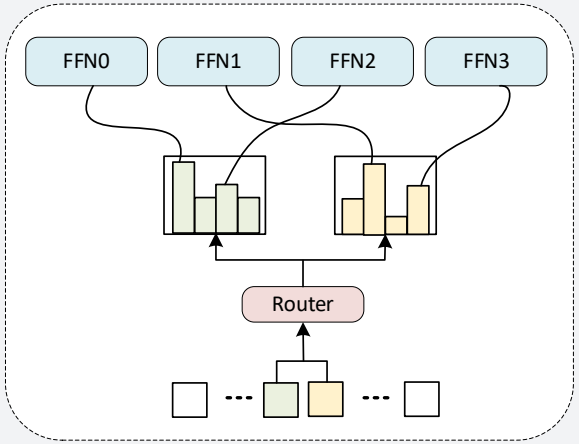
Type of Sparsity in ML Inference

MOFFETT AI

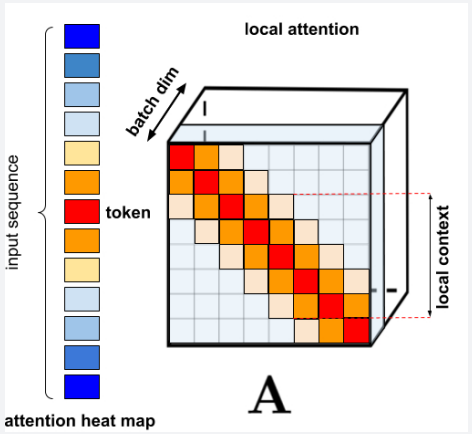
- **Static and Dynamic Sparsity**
  - **Static Sparsity**
    - Static Weight Sparsity (Pruning)
  - **Dynamic Sparsity**
    - Activation Sparsity
    - Conditional Sparsity
      - Contextual/Attention Sparsity
      - Mixture of Experts (MoE)
- **Sparsity Granularity**
  - **Coarse-granularity Sparsity**
  - **Fine-grained Sparsity**
- **Sparsity Patten**
  - **Structured sparsity**
  - Unstructured sparsity



Activation Sparsity



Mixture of Experts (MoE)

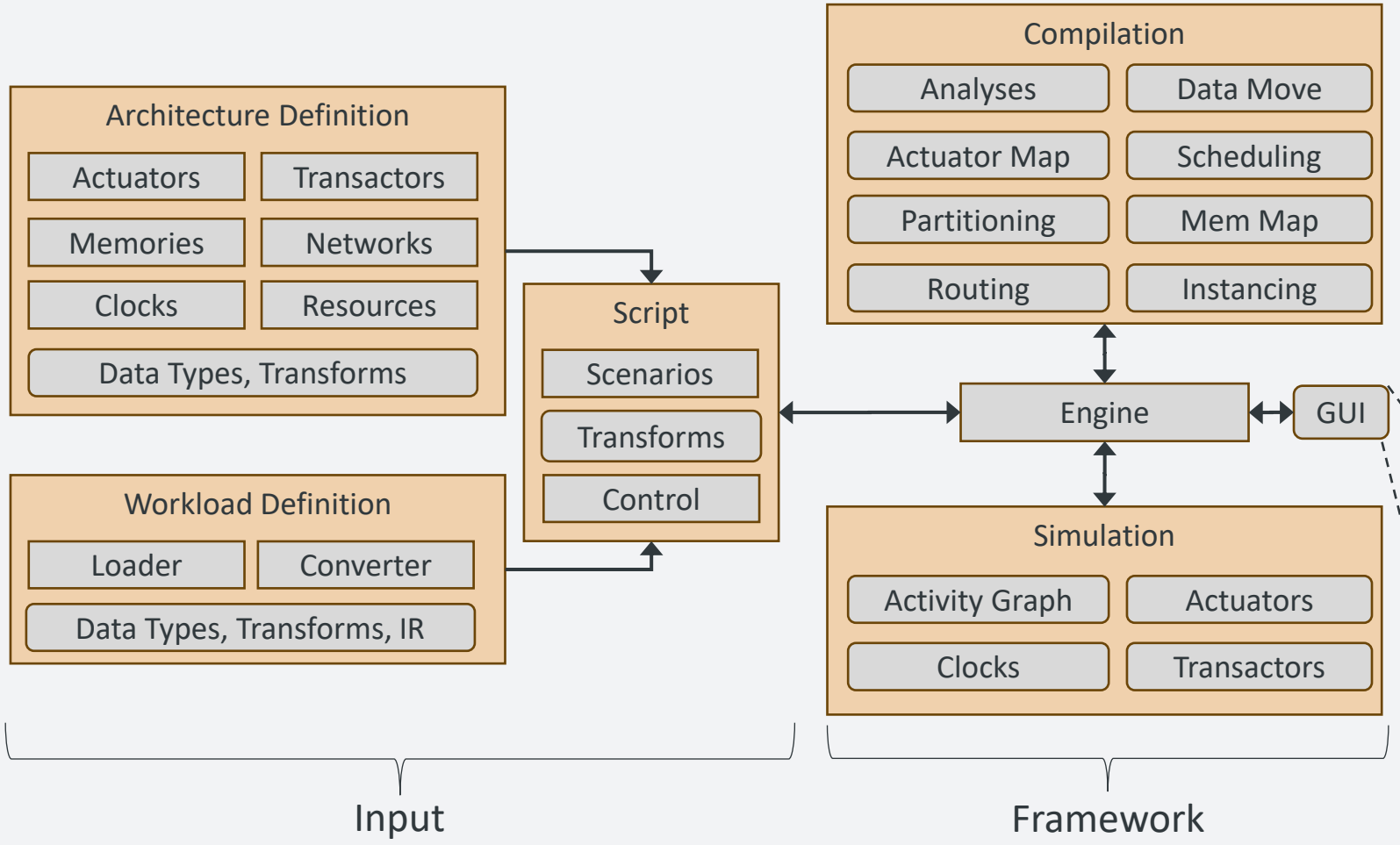


Sparse Attention

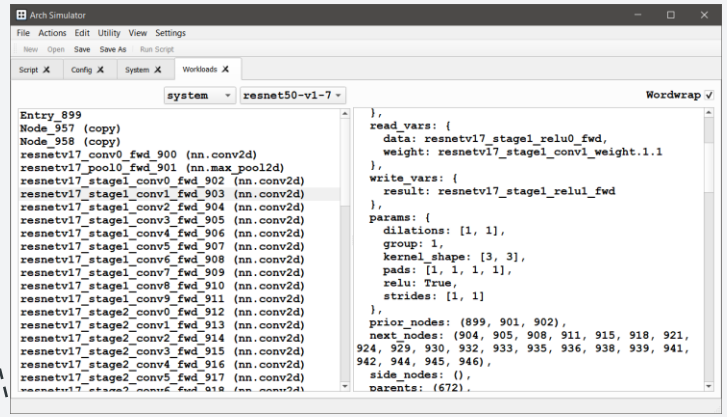
Conditional Sparsity

End-to-End Compile-Aware Architecture Simulator

MOFFETT AI



+ Rapid architectural exploration
  - Hardware Architecture Models
  - Workload Models
+ Adaptive compiler and simulator
+ Sparsity ratio and overhead in the loop
+ Results in seconds



Outline

- ML Trends and Antoum® Chip Overview
- Sparsity in ML Inference and Design Methodology
- **Antoum® Chip uArch and SparseOne Cards**
- Software Toolchain
- Application Performance and Demo

Overview of Antoum® Chip SoC Architecture

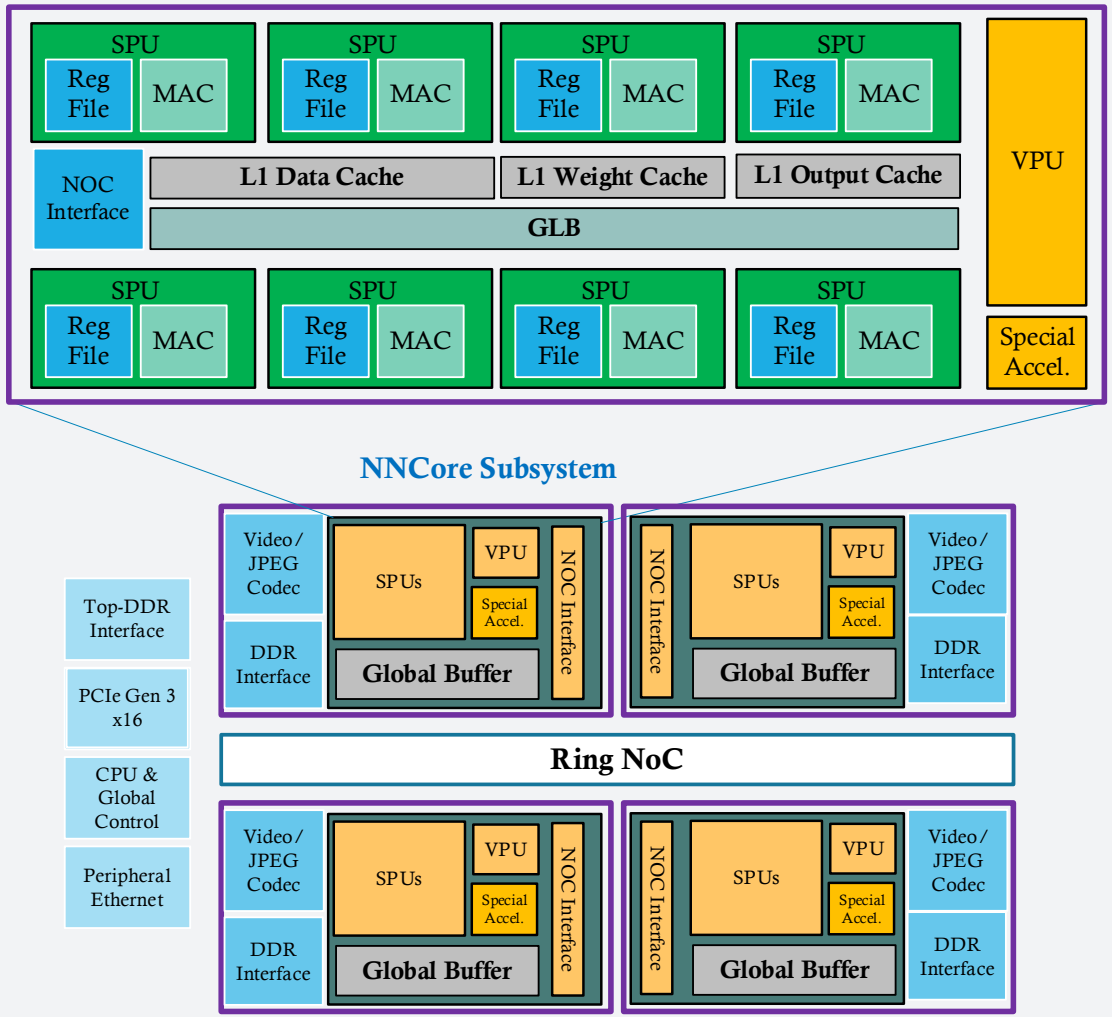
MOFFETT AI

Full SoC

- 4 x NNCore Subsystems
- PCIe Gen3 x 16
- 5 x LPDDR4x (20 GB)
- Quad-core ARM CPU (Linux)

NNCore Subsystem

- 8 x SPUs (Sparse Processing Units)
- 1 x VPU (512-element vector length)
- 1 x Scalar CPU Core
- Domain-specific Accelerators
  - Activation Function Accelerator (ACT)
  - Top-K Sorting Accelerator (TOPK)
  - Transpose Engine Accelerator (TRANS)
  - Global Copy Engine (Ring-SoC) (GLBCPY)
- Multi-level Memory System (Reg/L1/GLB)
- Multimedia System (Video/Image/PP Block)

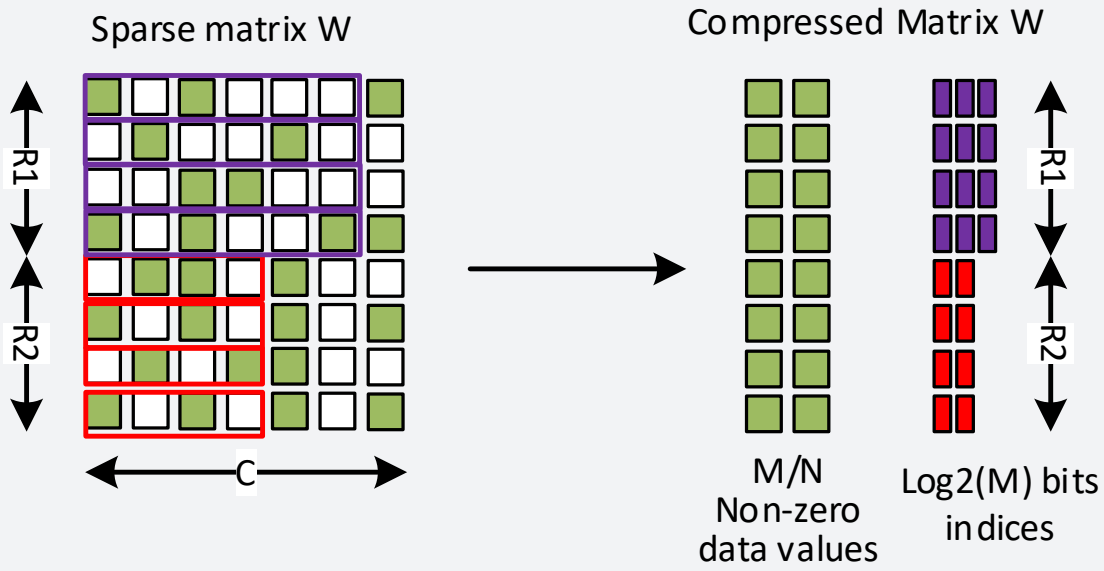


A Domain-specific Architecture with Deep-Sparsity Support

NNCore Subsystem: SPU

MOFFETT AI

- **Antoum SPU: N:M** Bank-balanced Structured Sparsity
  - Sparsity Ratio N = 1 to 32
  - Bank Size M = 16, 32, 64
  - Precision: INT8 and BF16
  - Both N, M and precision are programmable through commands
  - Supports native convolution and deconvolution and matrix multiplication
  - Supports dense convolution/matmul without extra memory overhead

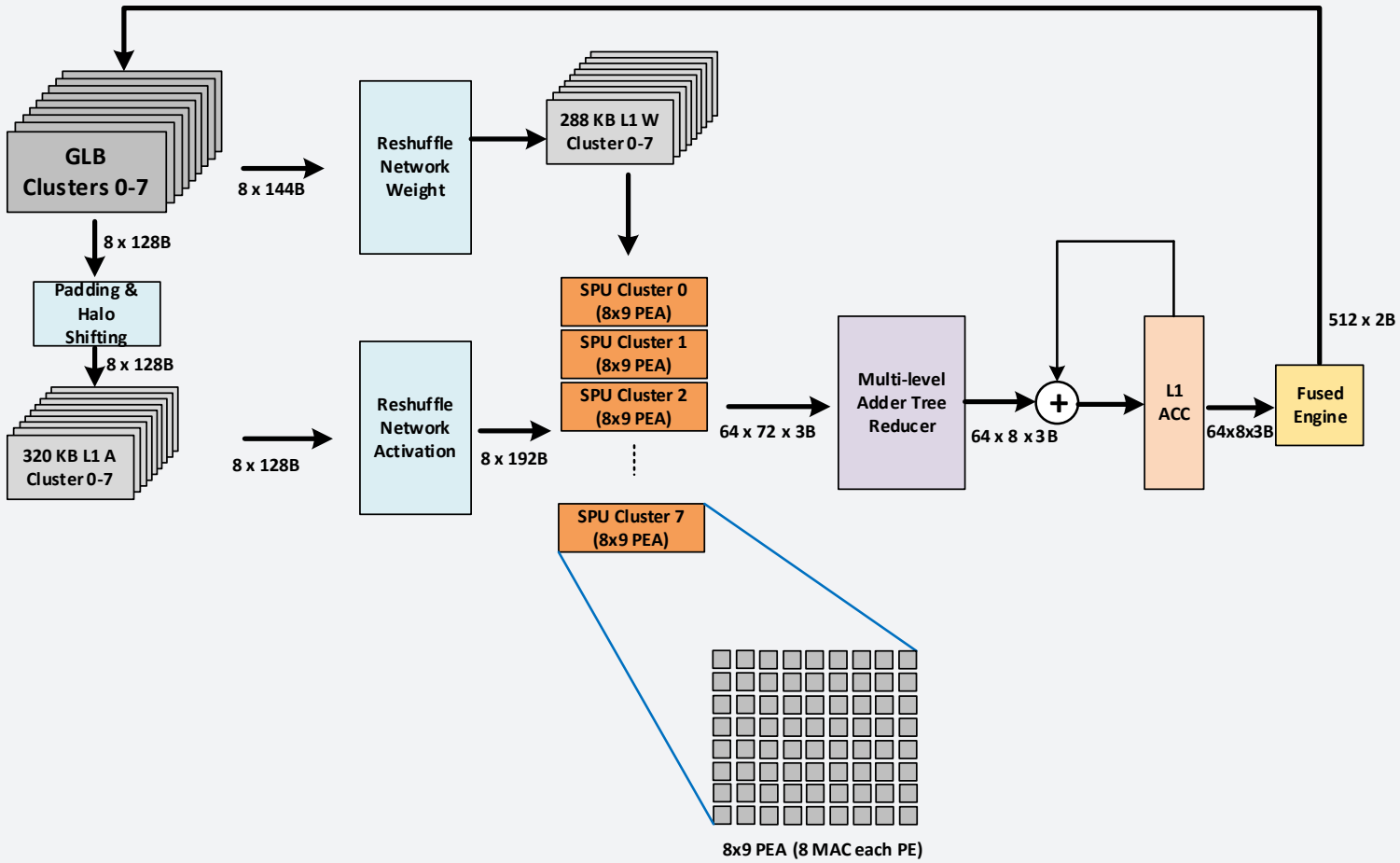


A Programmable (N, M) Bank-balanced Structured Sparsity matrix W, and its compressed representation
R1, R2 are other dimensions of the W tensor (N, K)

NNCore Subsystem: SPU Clusters Datapath

MOFFETT AI

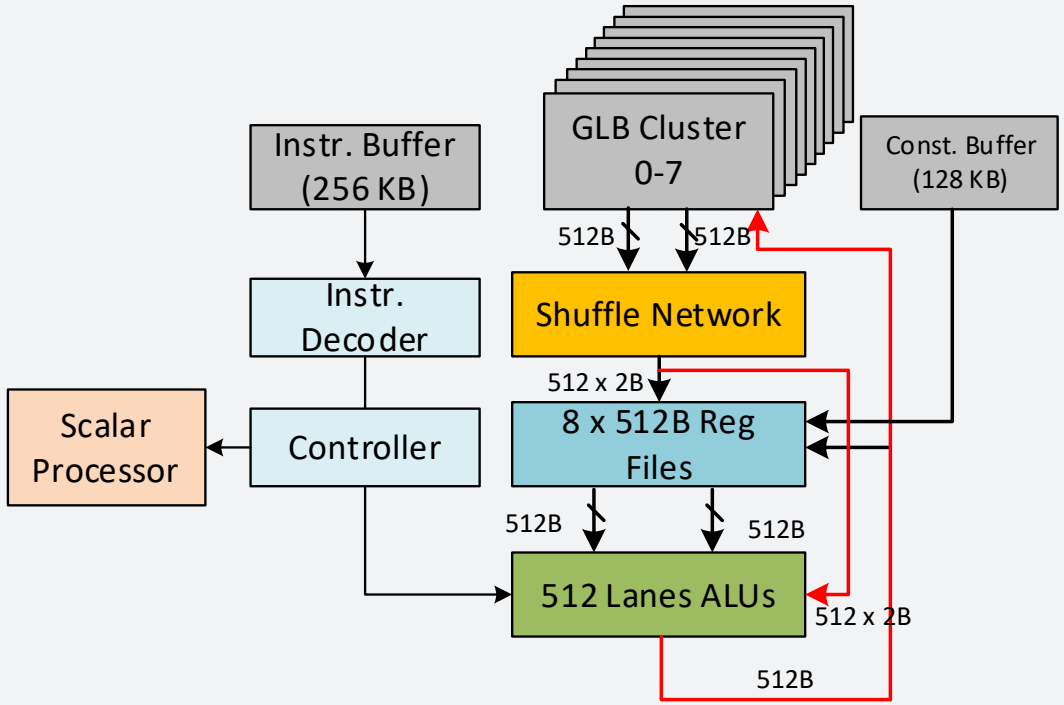
- **Harness both Spatial and Temporal parallelism**
  - 8 SPU clusters with 72 PEs per cluster and 8-lane MACs per PE
  - 20.5 MB Shared Global Buffer
  - 288 KB L1 Weight Buffer and 320 KB L1 Activation Buffer to maximize data reuse
  - Shuffle Network to support different weight and activation communication pattern (unicast, multicast and broadcast)
  - Native support of different-shaped convolution, deconvolution, matrix multiplication operations
  - Support various Fused Operations (BN, Elementwise OPs, Quantization, Activation Functions)



NNCore Subsystem: Vector Processing Unit (VPU)

MOFFETT AI

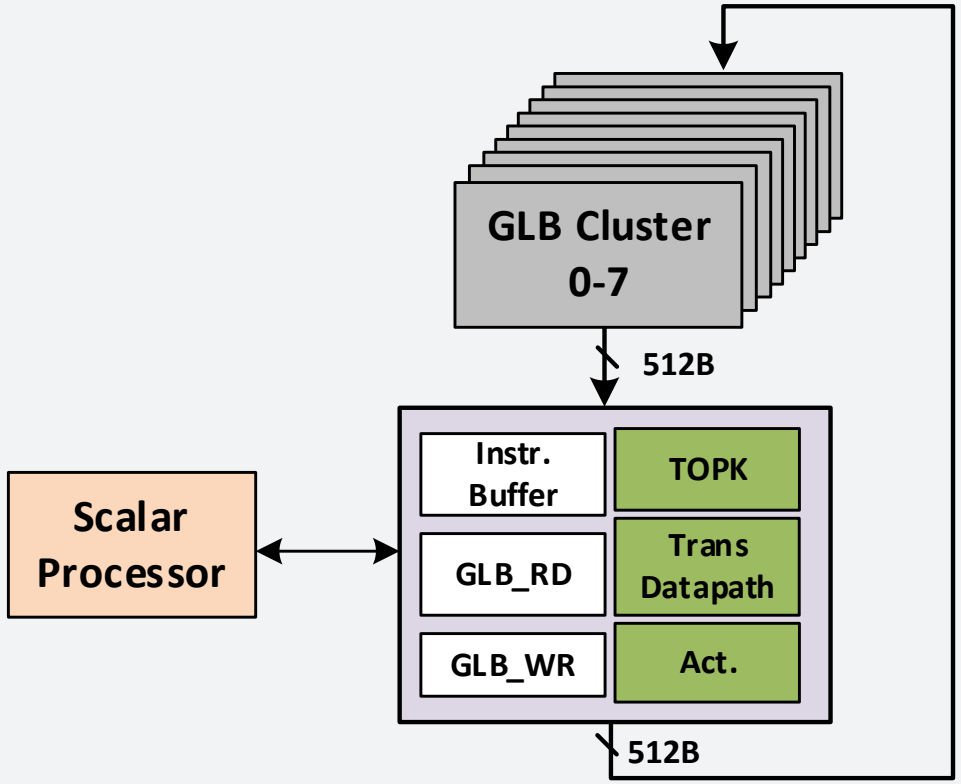
- Customized 512-lane Vector Processing Unit
  - Programmable (Pooling, Mean, Average, Element-wise Operations, etc)
  - Support INT8 and BF16 precisions
- 50+ 32-bit Instructions
  - RISC-like instructions
    - Config Instructions
    - Load/Store Instructions
    - Compute Instructions
  - CISC-like instruction
    - Operands from or to GLB/Const Buffers
- Shuffle Network
  - Broadcast, Shift and Masked data from/to GLB



NNCore Subsystem: Domain-Specific Accelerators

MOFFETT AI

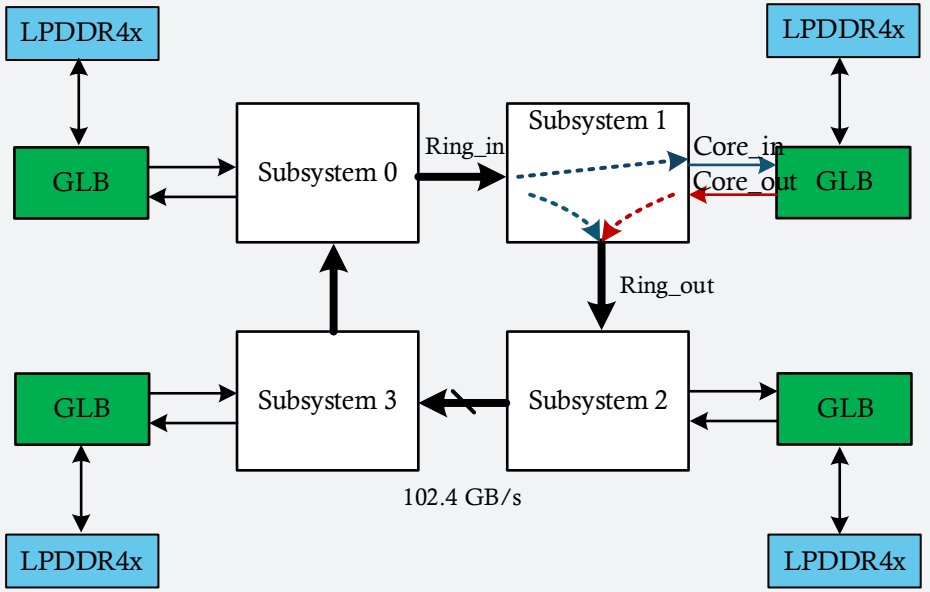
+ **Efficient Domain-specific Accelerators**
  - **ACT Engine**
    - Polynomial approximation of 10 different common operations (Exponential, Tanh, GeLU, Swish, Reciprocal, Reciprocal Square Root, etc)
  - **TRANS Engine**
    - Flexible transpose engine that can swap arbitrary tensor dimensions (N, H, W, C)
  - **TOPK Engine**
    - Iterative sorting algorithm to select top K largest number out of N numbers (N up to 32768, K up to 8192)



Antoum® Core 2 Core Interconnection

MOFFETT AI

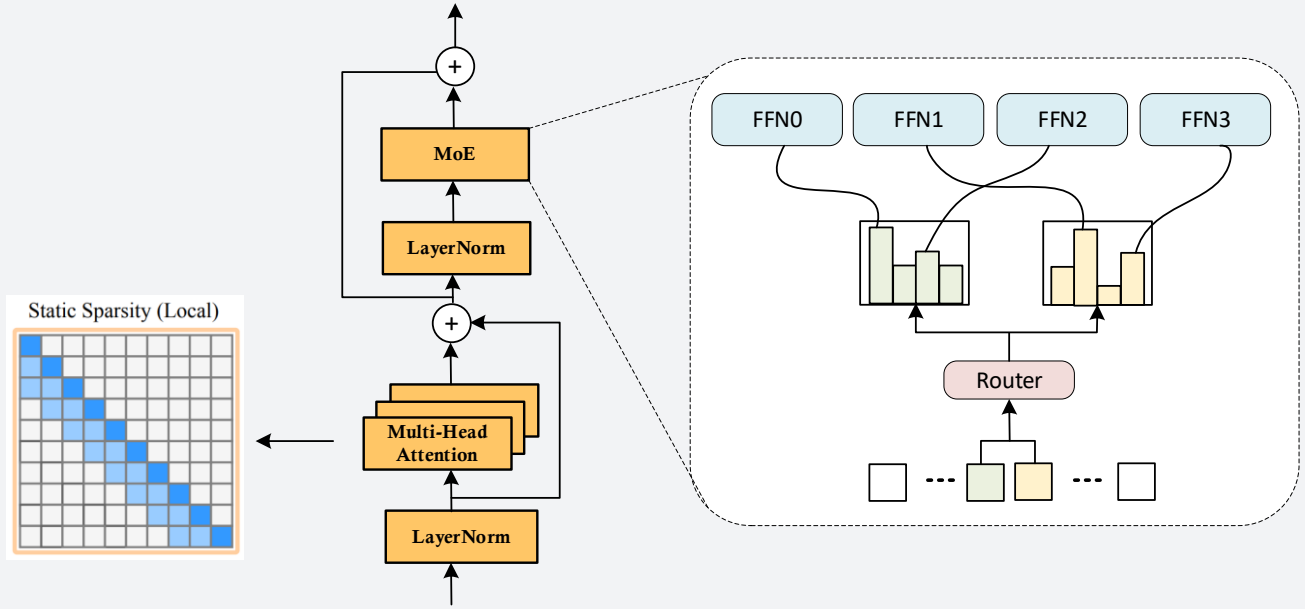
- **Core-2-Core Interconnection**
  - 115.2 GB/s Uni-directional Ring Interconnection
  - Supports Send, Receive, Sync programming primitives
- **Multi-core Communication Typical Use Cases**
  - Weight Sharing among NNCore subsystems
  - Data parallelism
  - Model parallelism (Pipelining)



Accelerate LLM on Antoum® : Hybrid Sparsity Support

MOFFETT AI

- **Sparse LLM is an active research area**
  - Mix of Expert (MoE) (E.g., switch transformers)
  - Contextual Sparsity
  - Sparse Attention (Fixed or dynamic pattern)
- **A hardware/software hybrid sparsity approach**
  - All static weight matrix can be pre-pruned with bank-balanced weight sparsity (FFN layers and QKV weight matrices)
  - Accelerators calculate the dynamic conditions for router layers (MoE), weight block selectors (Contextual Sparsity)
  - Scalar CPU core schedules which expert or block to execute based on pre-calculated conditions



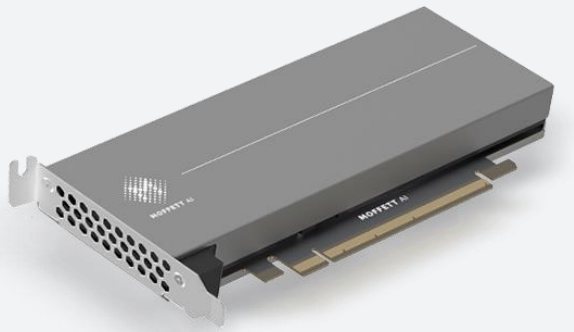
Moffett Antoum® Chip Spec

Key Spec	Numbers
Area	• 340 mm^2 @ 12 nm
Frequency	• 800 MHz
Peak Performance Sparse Processing Units (SPU)	• 29.5 TOPS @ INT8 (up to 32x boost via sparsity) • 14.7 TFLOPS @ BF16 (up to 32x boost via sparsity)
Peak Performance Vector Processing Unit (VPU)	• 3.7 TFOPS @ BF16
Host Interfaces	• PCIe Gen 3 x 16
Memory	• 20 GB LPDDR4x • 84 GB/s peak bandwidth
Video Codec	• 64-channel H.264 decoding 1080p @ 30 fps • 8-channel H.264 decoding 1080p @ 30 fps
JPEG Decoding	• JPEG decoding 1080p @ 1600 fps
TDP*	• 70 W

*TDP is based on a full sing-chip PCIe card, chip power is lower

Moffett SparseOne® AI Inference Cards

MOFFETT AI

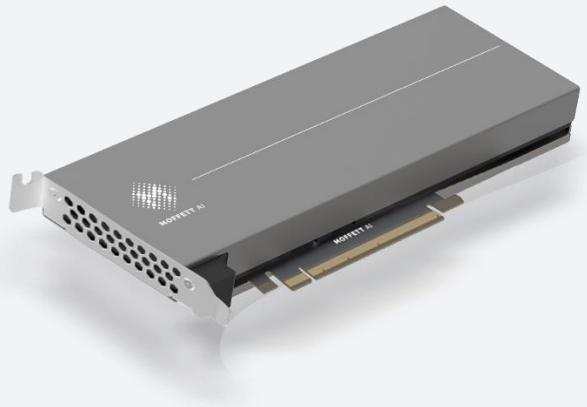


SparseOne® S4

Performance*:

- 29.5 TOPS @ INT8
- 18.4 TFLOPS @ BF16

Interfaces: PCIe-Gen3 x 16
Memory: 20 GB LPDDR4x
TDP: 70 W

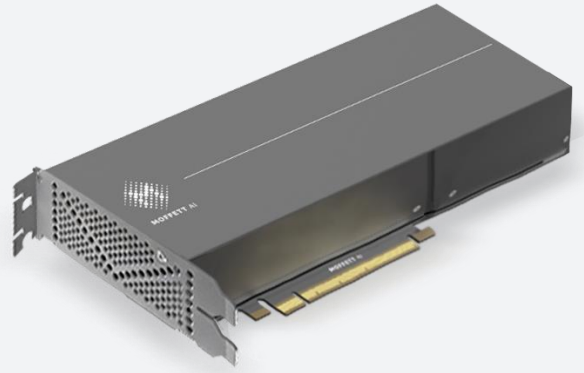


SparseOne® S10

Performance*:

- 59.0 TOPS @ INT8
- 36.8 TFLOPS @ BF16

Interfaces: PCIe-Gen4 x 16
Memory: 40 GB LPDDR4x
TDP: 150 W



SparseOne® S30

Performance*:

- 88.5 TOPS @ INT8
- 55.2 TFLOPS @ BF16

Interfaces: PCIe-Gen4 x 16
Memory: 60 GB LPDDR4x
TDP: 250 W

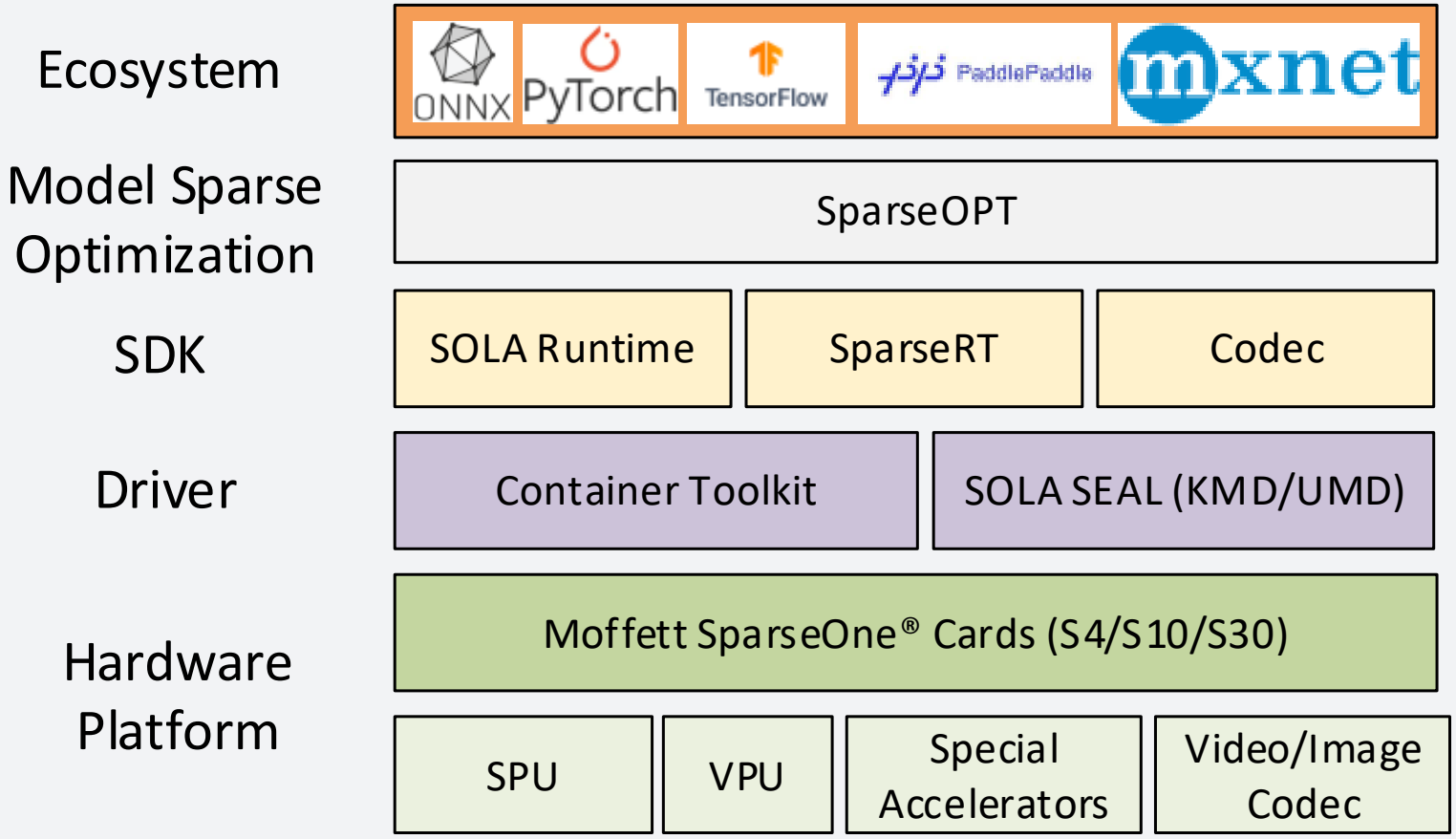
*Peak dense performance, up to 32x boost via sparsity

Outline

- ML Trends and Antoum® Chip Overview
- Deep-Sparsity Hardware Software Co-design
- Antoum® Chip uArch and SparseOne Cards
- **Software Toolchain**
- Application Performance and Demo

SparseOne® Software Toolchain

MOFFETT AI



SparseOne® Software Toolchain Workflow

MOFFETT AI

Step 1. SparseRT converts models to a unified format (Moffett IR)

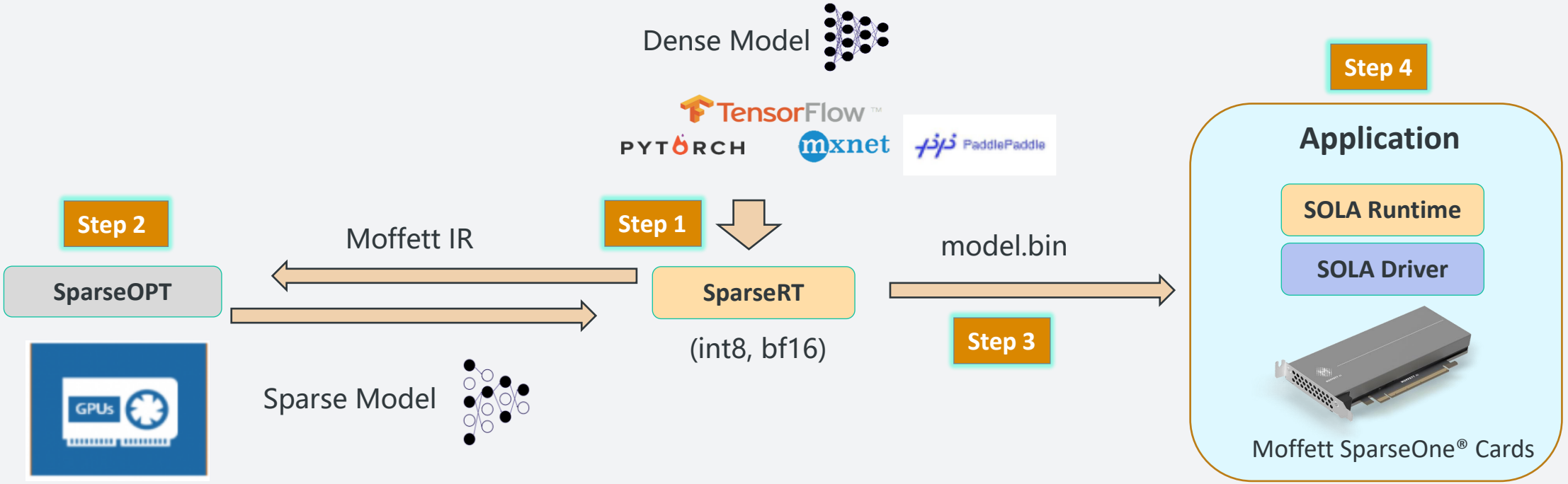
Step 2. SparseOPT sparsifies and quantizes models

Step 3. SparseRT compiles sparsified models to binary format

Step 4. Load & execute models on HW with SOLA Runtime.

- INPUT:
  - resnet50_ir
- OUTPUT:
  - resnet50_ir_optimized

```python
import nn_optimizer
...
pth_model = nn_optimizer.pytorch_model(
    './examples/resnet50/resnet50_ir')
...
# training loop
for epoch in range(n_epochs):
    # user's training code here
    ......
    optimizer.step()
    # perform model efficiency optimization
    nn_optimizer.opt_step()
    ......
```



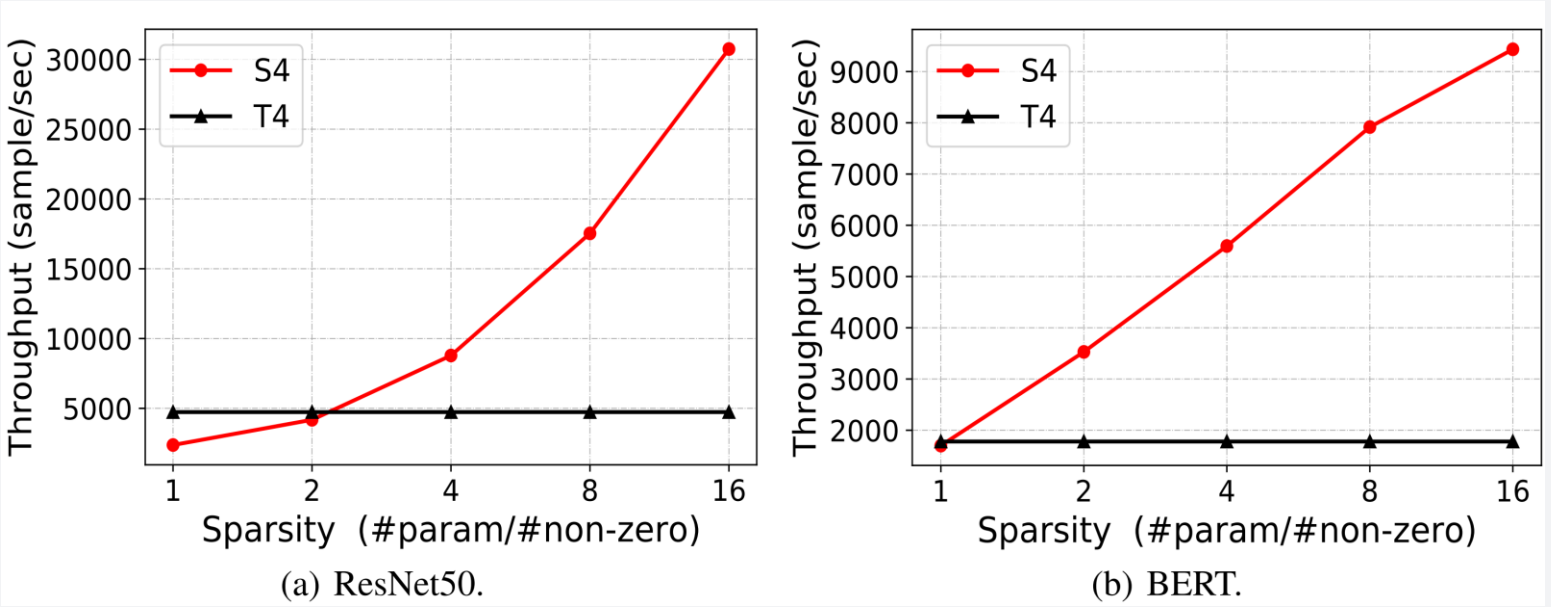
Outline

- ML Model Trends and Architecture Design Methodology
- Deep-Sparsity Hardware Software Co-design
- Antoum® Chip uArch and SparseOne Cards
- Software Toolchain
- **Application Performance and Demo**

Accuracy, Throughput and Sparsity Ratio Trade-off

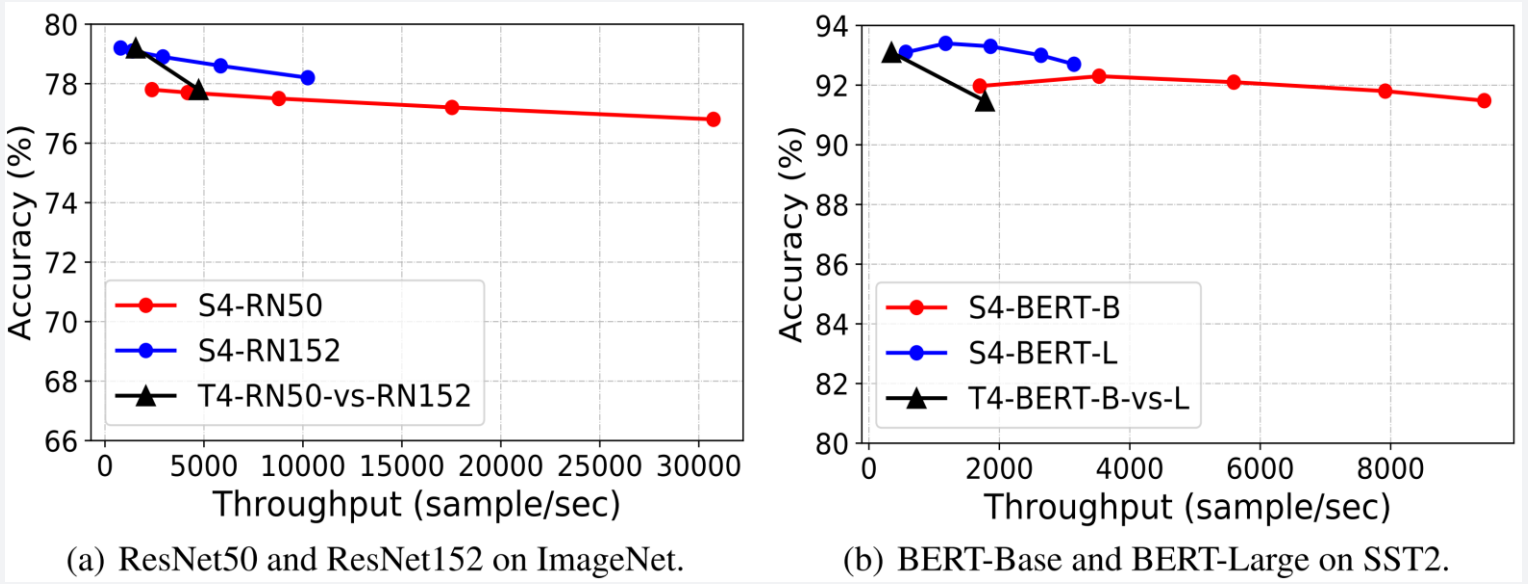
MOFFETT AI

Throughput vs Sparsity
ResNet50 and BERT-Base



(a) ResNet50. (b) BERT.

Accuracy vs Throughput
ResNet and BERT

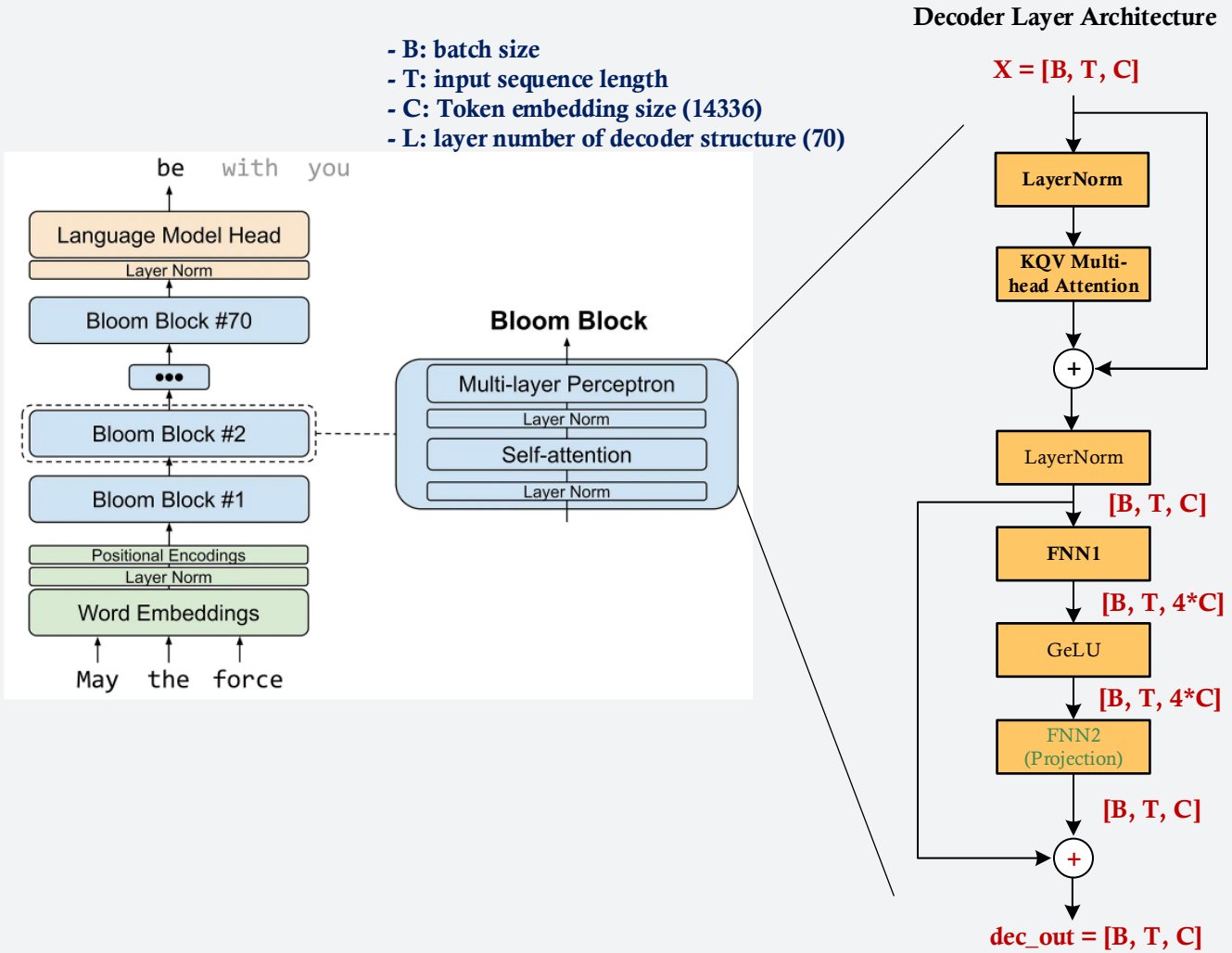


(a) ResNet50 and ResNet152 on ImageNet. (b) BERT-Base and BERT-Large on SST2.

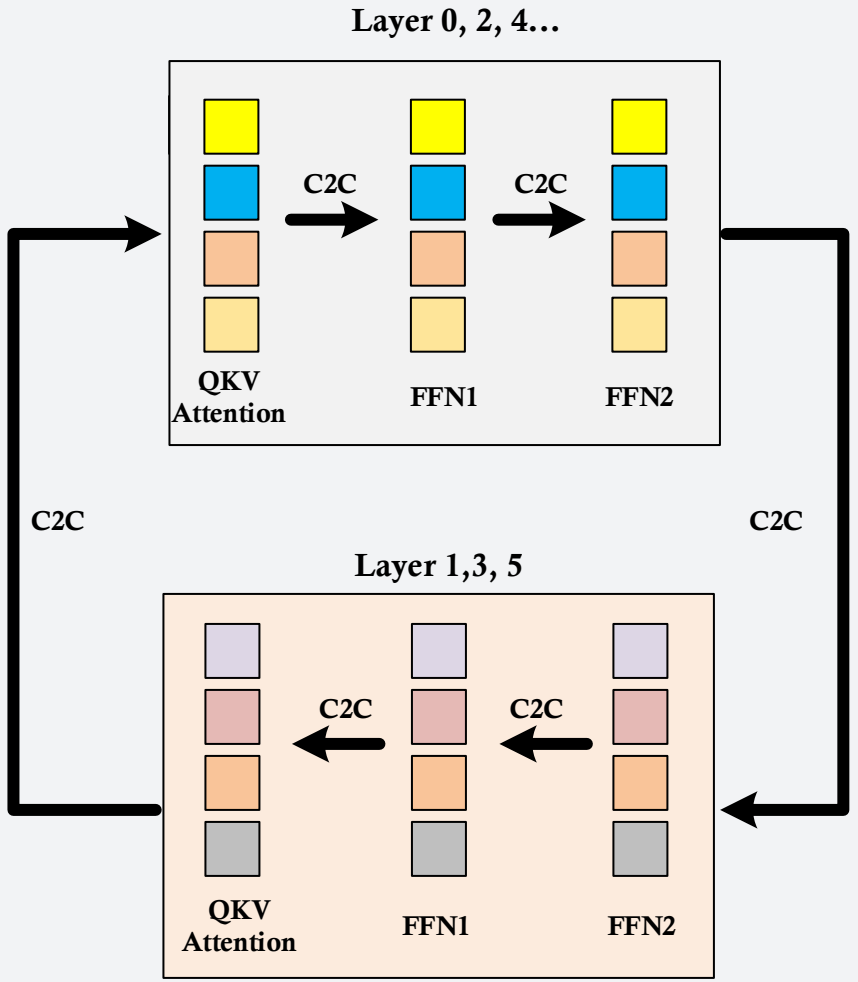
HOT CHIPS

Multi-card System to Support Bloom-176B (8 x S30)

MOFFETT AI

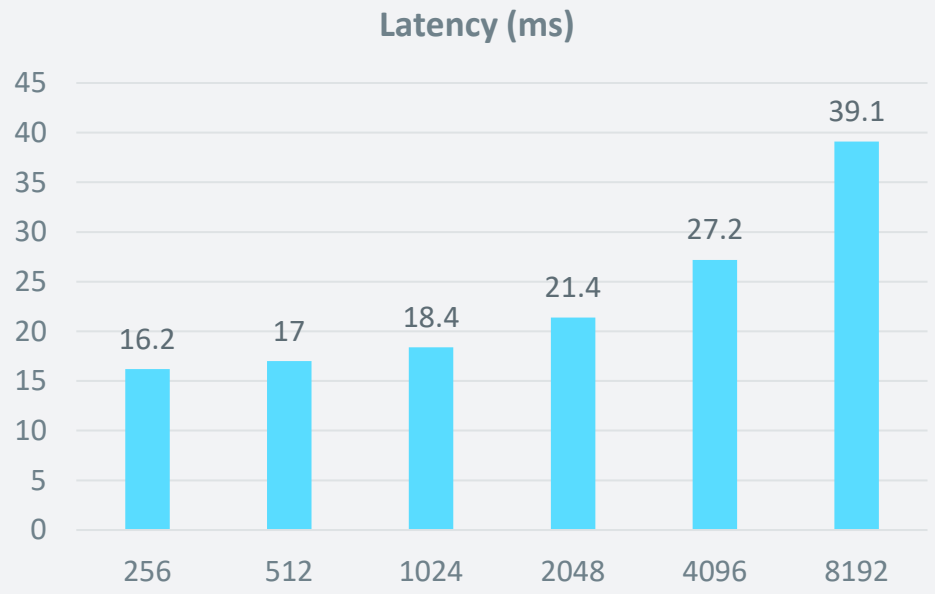


BLOOM-176B Model Architecture

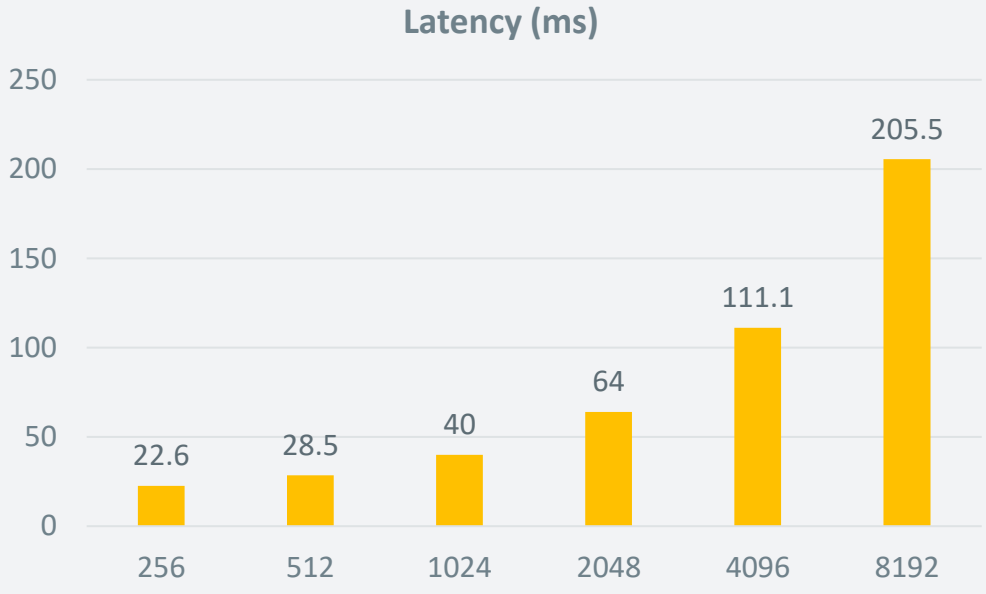


Model Mapping to 8 x S30

Bloom-176B on 8 x Moffett S30



Batch Size = 1
Context Length = 256 to 8192



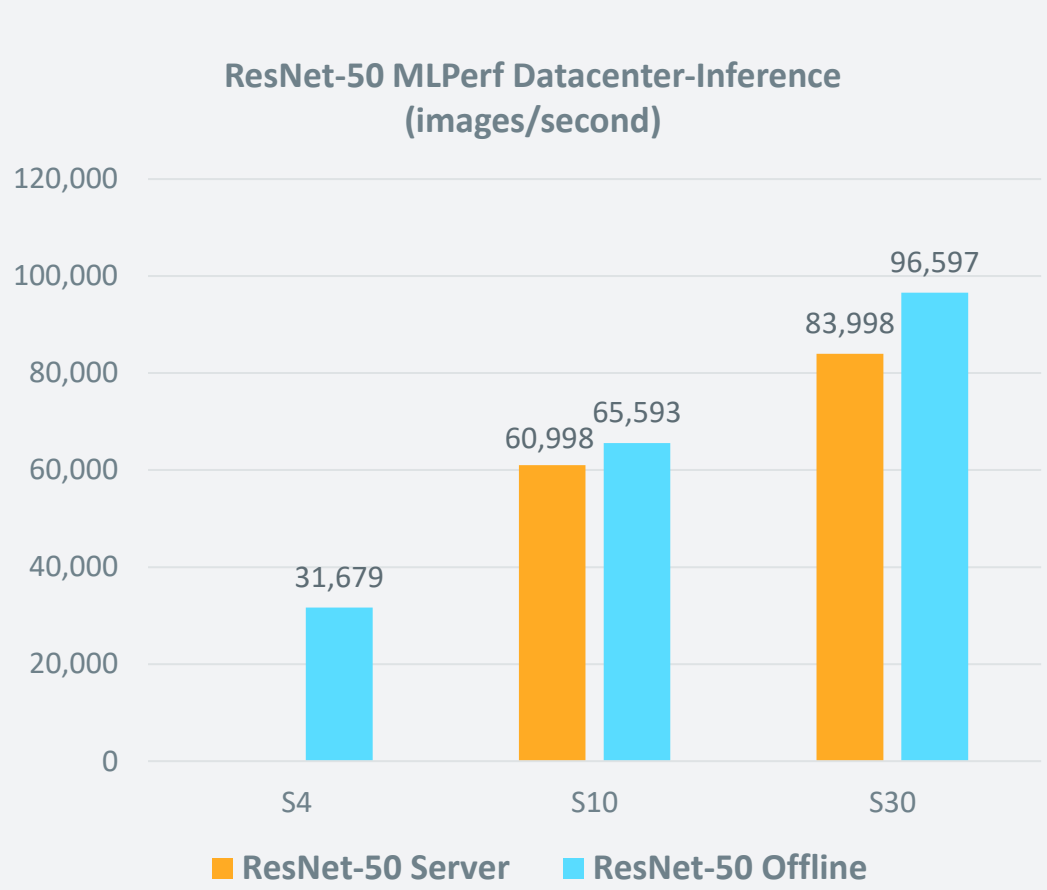
Batch Size = 8
Context Length = 256 to 8192

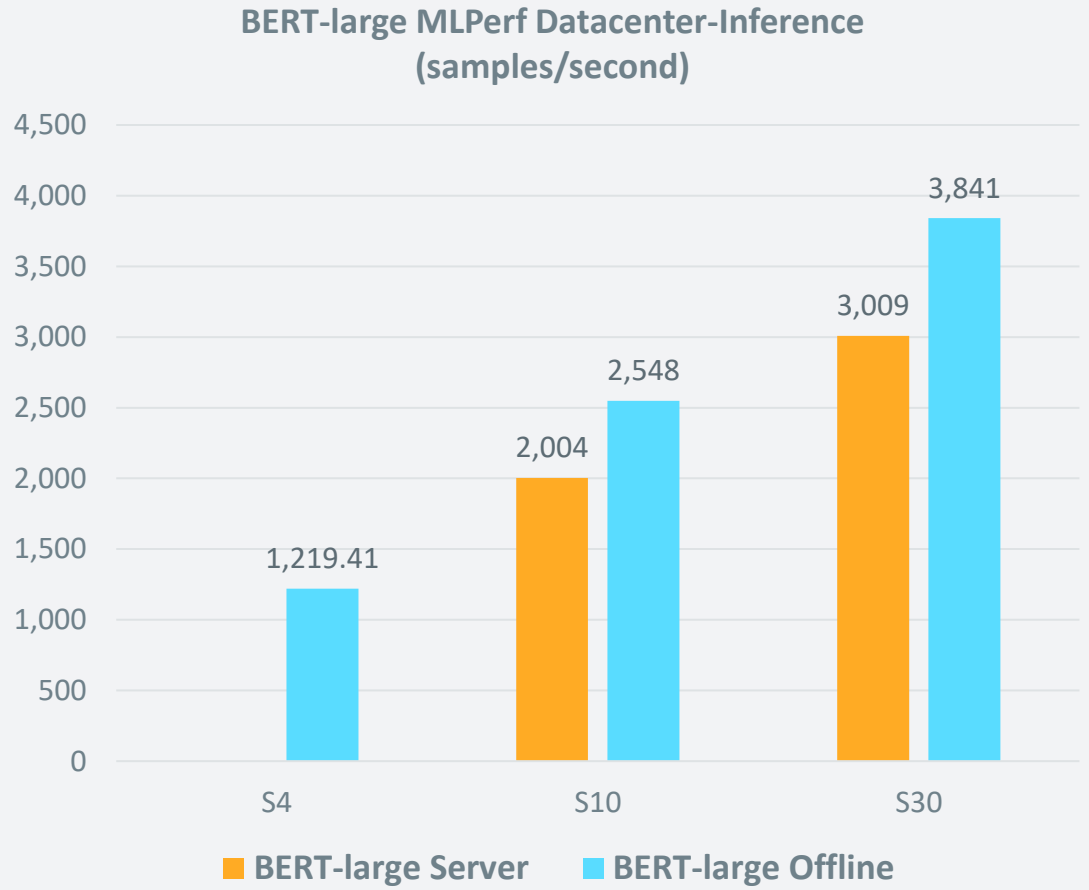
32x Weight Sparsity with Hybrid INT8/BF16 Performance

SparseOne® Cards MLPerf Benchmark Results

MOFFETT AI

- MLPerf v2.1 and v3.0 Data-center Inferences Results (Open Division)
- MLPerf v3.1 GPT-J-6B results will be announced soon





A Demo: Sparse Super Resolution on S4

MOFFETT AI

Model Stat.

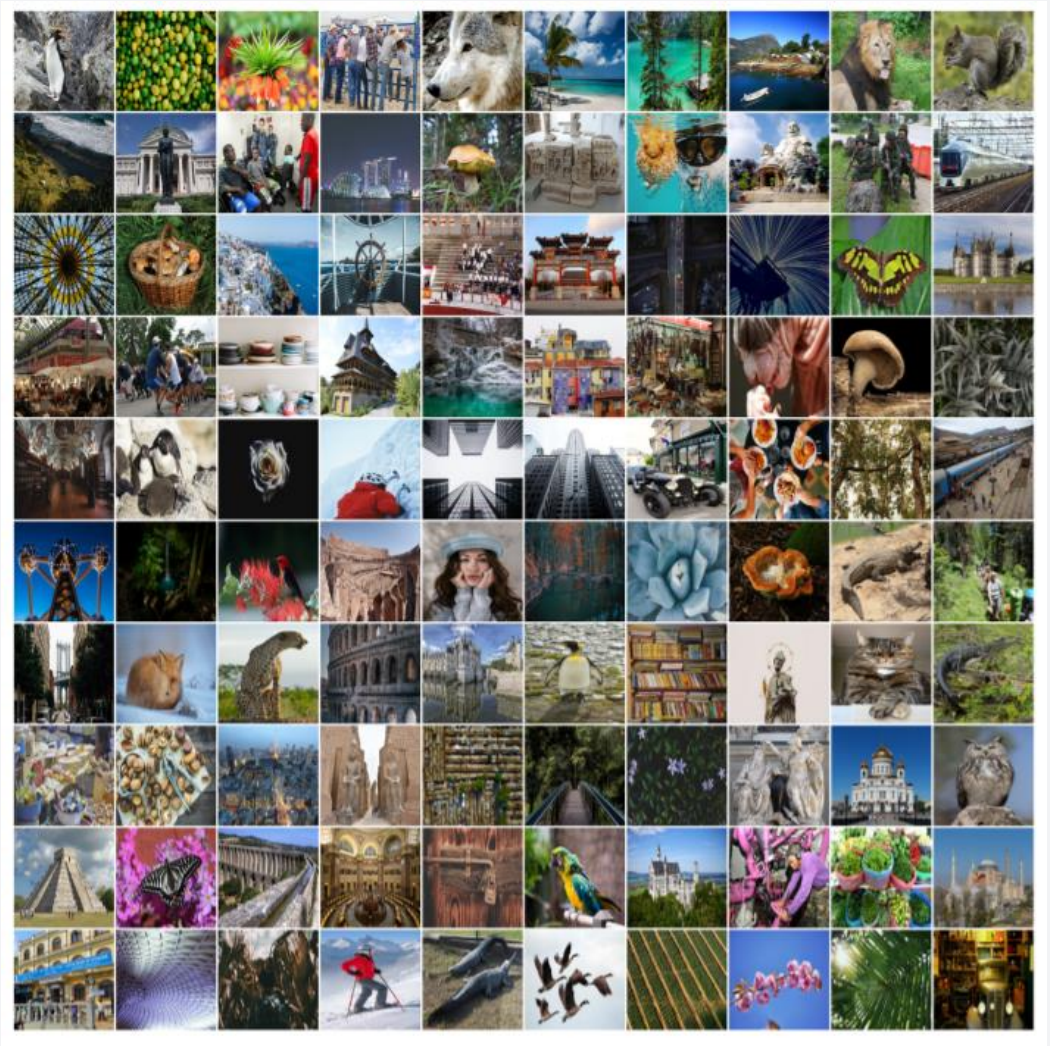
Model	Input Size	Output Size	GFLOP	MParam
RCAN (8x5)	(360, 640)	(720, 1280)	766	3.34

- Change from original ECCV18 RCAN (20x10 units).

Data Stat.

Dataset	resolution	Train #Images	Test #Images
DIV2K	2K	800	200

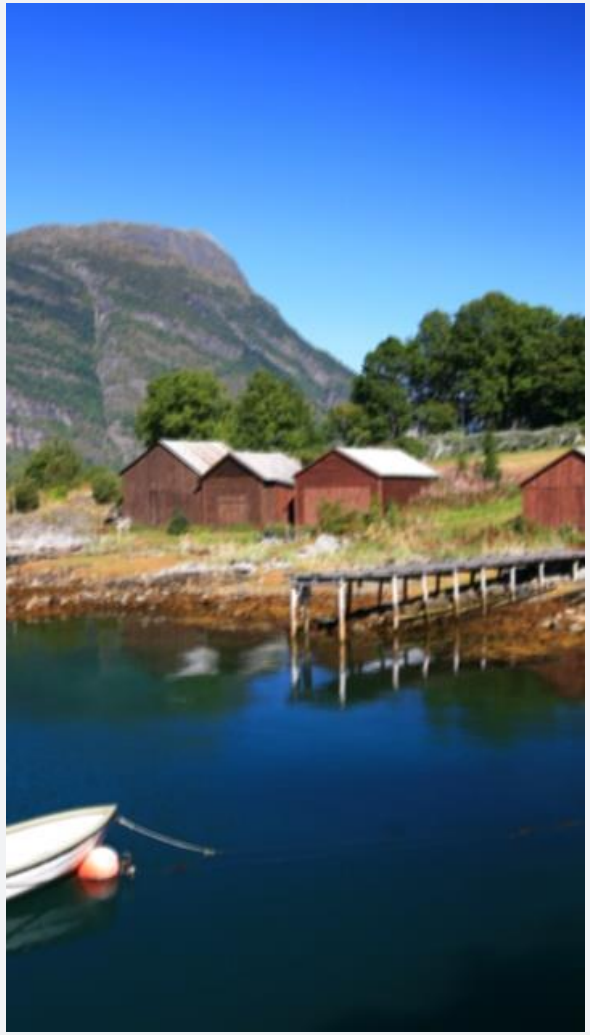
- Dataset used for challenges in NTIRE (New Trends in Image Restoration and Enhancement) of CVPR 2018, ECCV 2018. (https://data.vision.ee.ethz.ch/cvl/ntire20/)



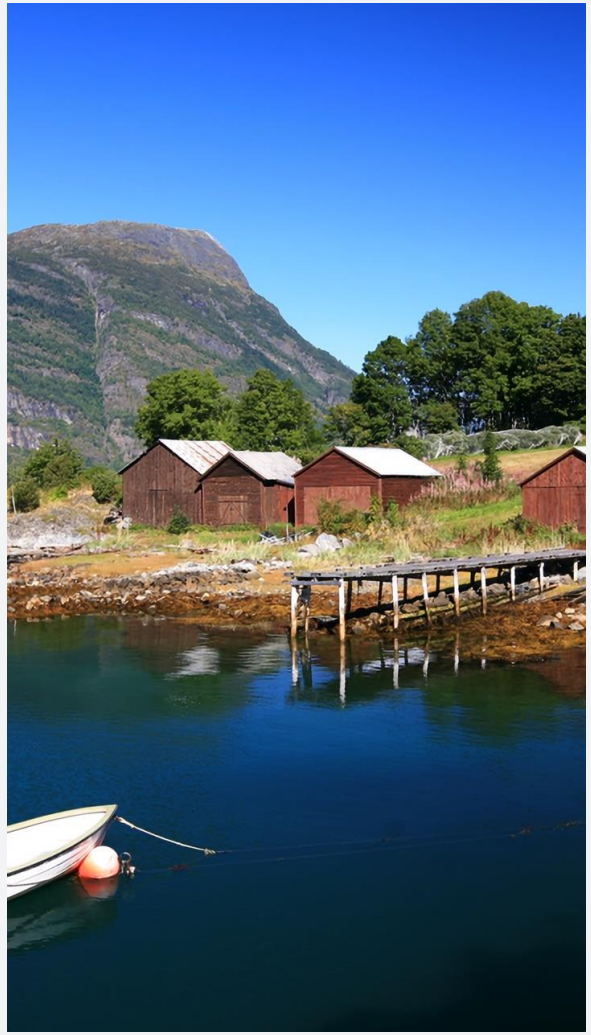
A Demo: Sparse Super Resolution on S4

MOFFETT AI

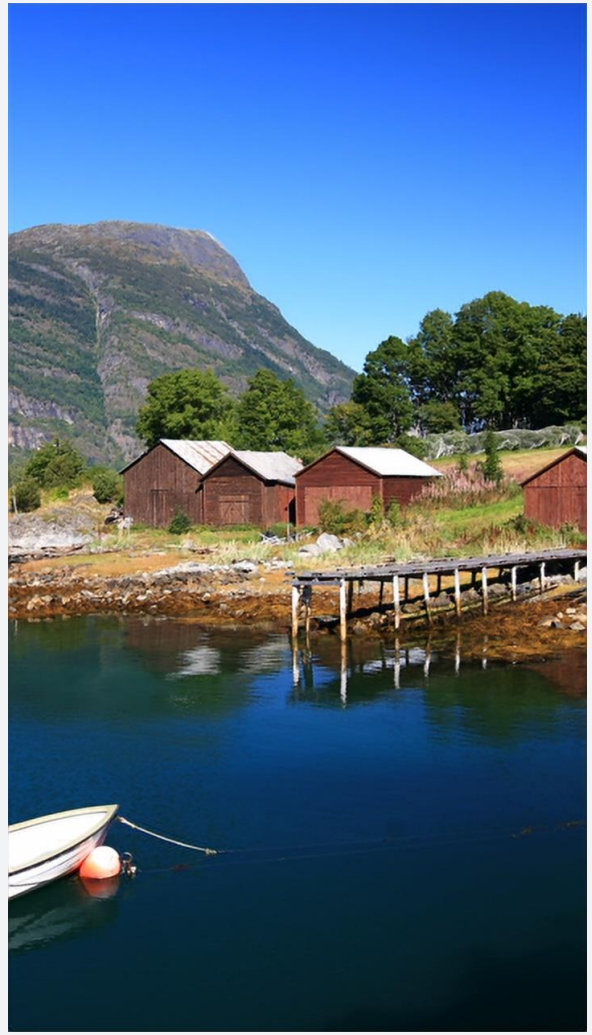
Input Image



Original Model
9 FPS



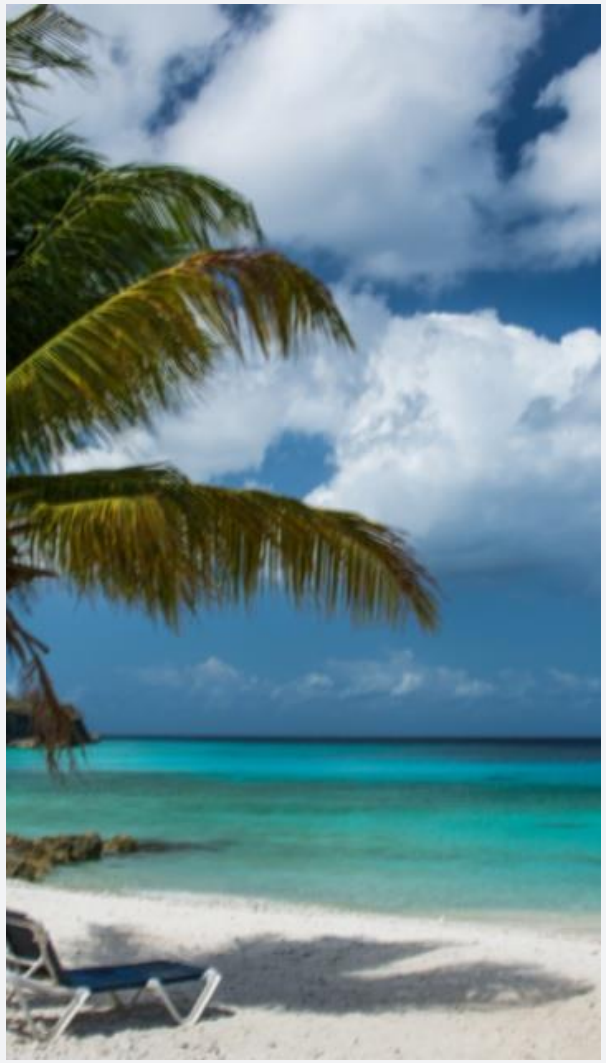
8x Compressed on S4
59 FPS



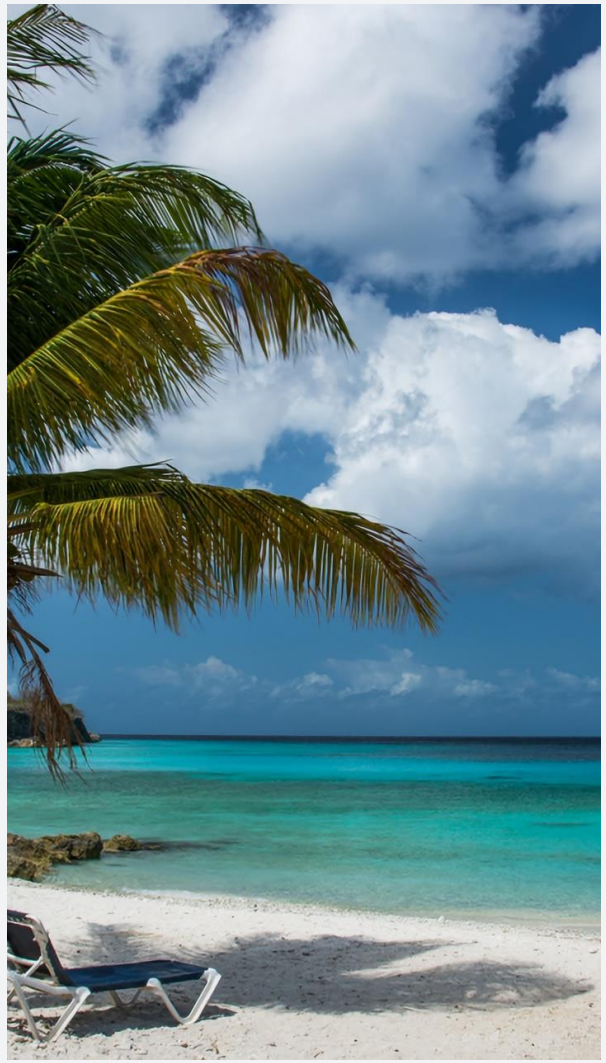
A Demo: Sparse Super Resolution on S4

MOFFETT AI

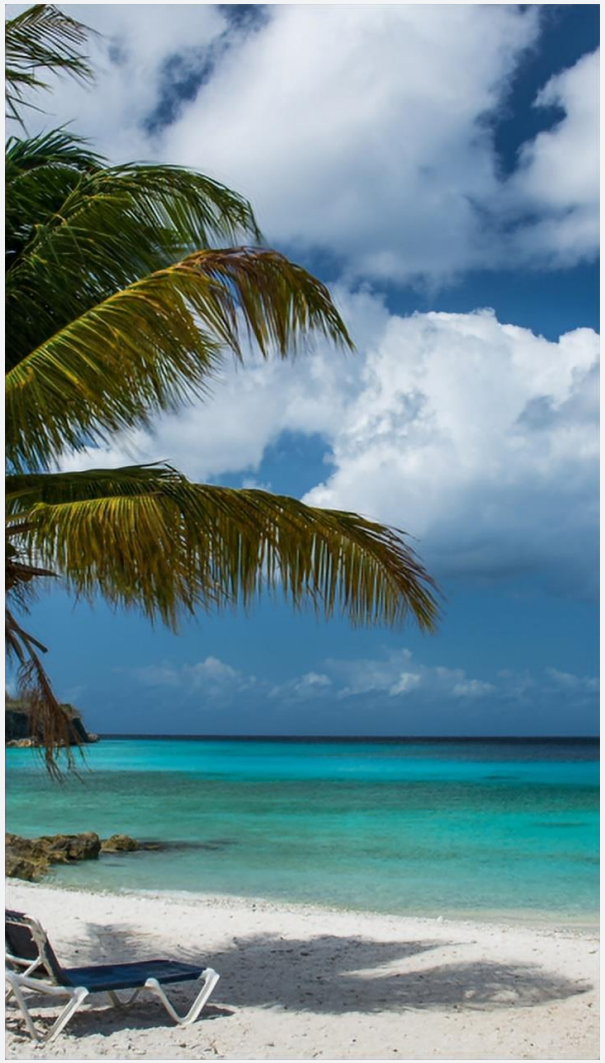
Input Image



Original Model
9 FPS



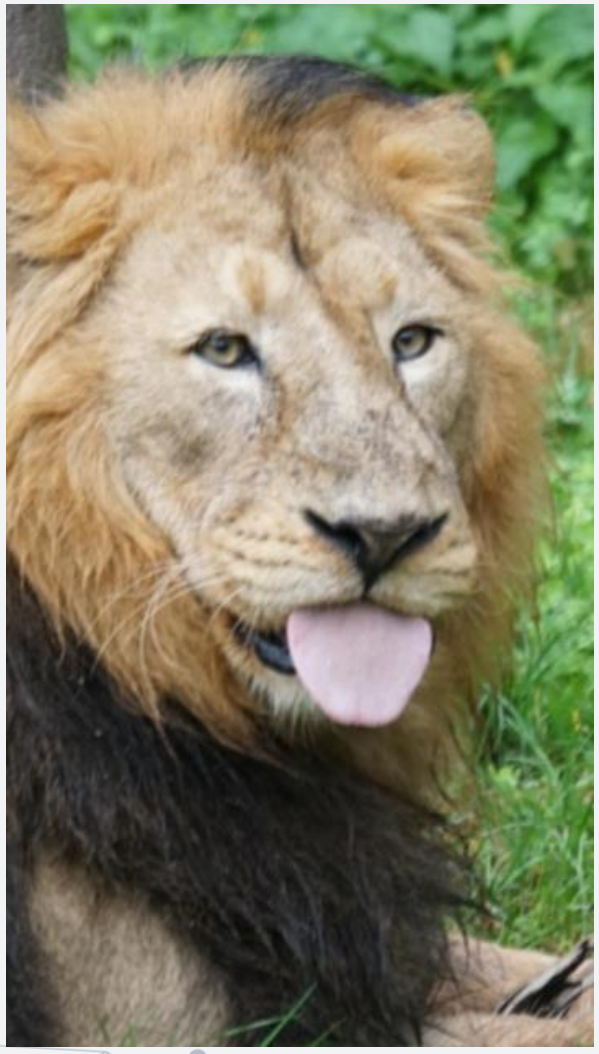
8x Compressed on S4
59 FPS



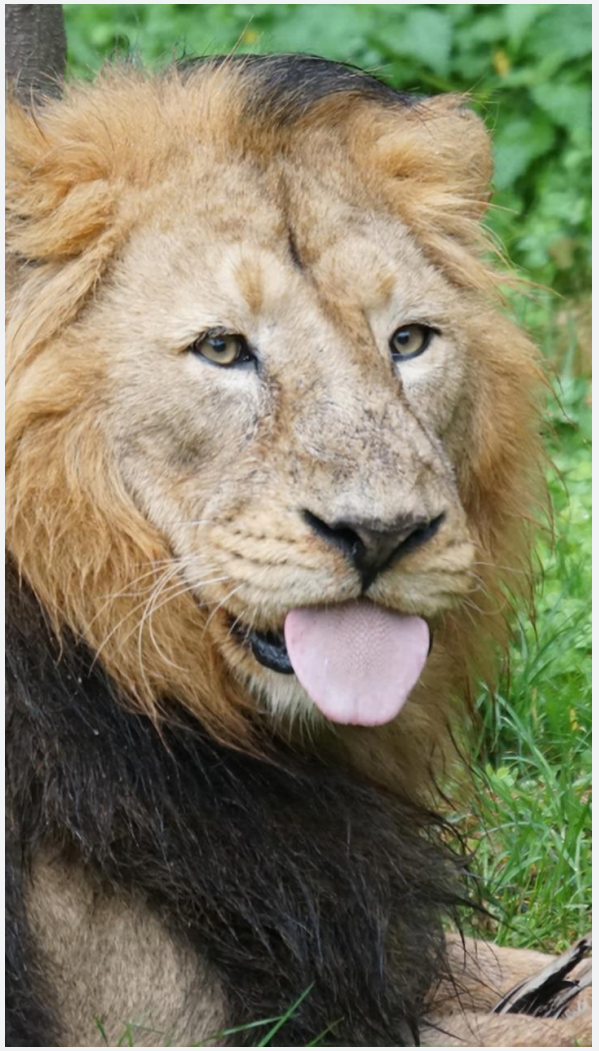
A Demo: Sparse Super Resolution on S4

MOFFETT AI

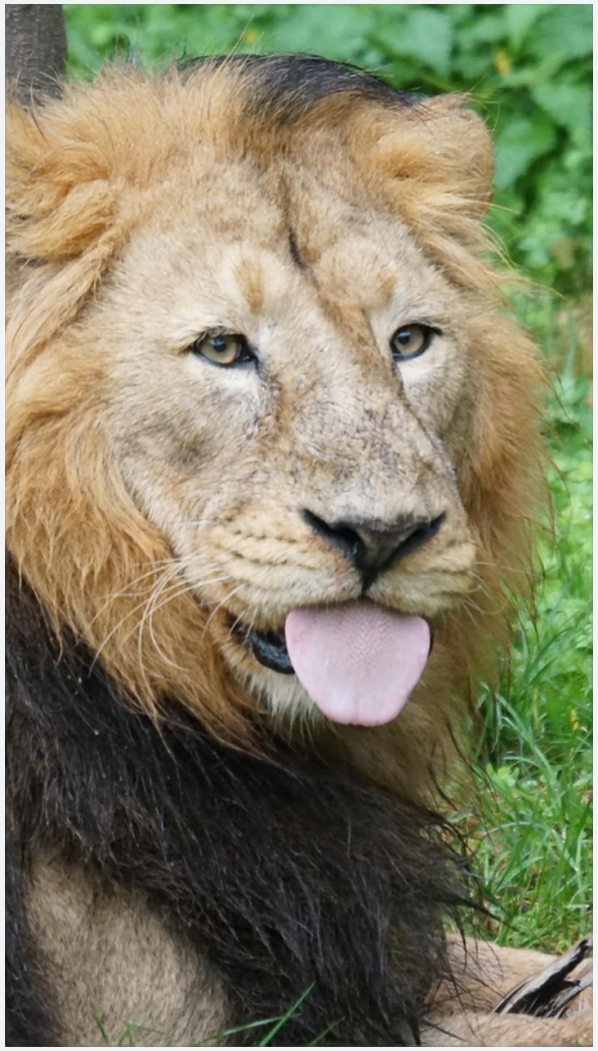
Input Image



Original Model
9 FPS



8x Compressed on S4
59 FPS



Summary

MOFFETT AI

- We introduce Antoum®, a data-center SoC chip for vision and large language model inference
  - Support flexible bank-balanced sparsity (up to 32x sparsity)
  - Near-memory architecture (82 MB Multi-level on-chip SRAM)
  - Scalable, Flexible and Efficient architecture
    - Multi-core design with ring interconnection
    - High-bandwidth vector processing unit
    - Various domain-specific accelerators
  - An end-to-end compile-aware architecture simulator for design space exploration
  - A complete software toolchain for efficient model deployment
  - Demonstrated results on MLPerf Benchmarks for both vision and language models

MOFFETT AI

Thank you and Questions?

HOT CHIPS